

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

Block and convolutional codes decoding method for digital communication

Patent number: DE19725275
Publication date: 1998-12-24
Inventor: HAGENAUER JOACHIM (DE)
Applicant: HAGENAUER JOACHIM PROF DR ING (DE)
Classification:
- **international:** H03M13/00; H03M13/12
- **european:** H03M13/45
Application number: DE19971025275 19970614
Priority number(s): DE19971025275 19970614

Abstract of DE19725275

The method involves using a decoder comprising a nonlinear network, which is derived from the code or parity equations, in which all bit or symbols in an equation are represented by their log-likelihood values as real values. In the network, all symbols connected by code or parity equations are connected by a boxplus element, while all symbols are stored in a circuit element and represented as currents, charges, or voltages. These values can also be represented as real fixed-point or floating-point variables, e.g. in a processor or circuit. Conversion tables and nonlinearities such as tangent hyperbolic and inverse tangent hyperbolic functions are also used.

Data supplied from the **esp@cenet** database - Worldwide



①⑨ **BUNDESREPUBLIK
DEUTSCHLAND**



**DEUTSCHES
PATENT- UND
MARKENAMT**

⑫ **Offenlegungsschrift**
⑩ **DE 197 25 275 A 1**

⑤① Int. Cl.⁶:
H 03 M 13/00
H 03 M 13/12

②① Aktenzeichen: 197 25 275.3
②② Anmeldetag: 14. 6. 97
④③ Offenlegungstag: 24. 12. 98

DE 197 25 275 A 1

- ⑦① Anmelder:
Hagenauer, Joachim, Prof. Dr.-Ing., 80333
München, DE
- ⑦② Erfinder:
Hagenauer, Joachim, 80333 München, DE
- ⑤⑥ Entgegenhaltungen:
B.Friedrichs:"Kanalcodierung", Springer Verlag
Berlin, 1995;
J. Hagenauer, E. Offer, L. Papke: "Iterative
decoding of binary block and convolutional codes",
IEEE Trans. on Inf. Theory, Vol.IT-42, S. 429-445,
March 1996;
J. Hagenauer:"Soft-In/Soft-Out: The benefits of
using soft decisions in all stages of digital
receivers", in: Proc. 3rd Int. Workshop on DSP
Techniques applied to Space Communications,
ESTEC
Noordwijk, The Netherlands, Sept. 1992;
C. Berrou et al:"Near Shannon limit error-correc-

ting and decoding: Turbo-Codes(1)", Proc. IEEE
International Conference on Communication (ICC),
Geneva, Switzerland, S.1064-1070, Mai 1993;
J. Lodge et al:"Separable MAP filters for the de-
coding of product and concatenated codes", Proc.
IEEE International Conference on Communication
(ICC), Geneva, Switzerland, S.1740-1745, Mai1993;
Th. Woerz, J. Hagenauer: "Decoding of M-PSK-
Multilevel Codes", European Transactions on Tel-
communications ETT, Vol.4, No.3, S. 299-308, 1
993; R.M. Tanner: "A recursive approach to low co
mple- xity codes", IEEE Transactions on Informat
ion Theory, Vol. IT-27, S. 533-547, Sept. 19
81; G. D. Forney: "The forward-backward al
gorithm", Proc. of the 1996 Allerton Conferenc
e, Allerton Illinois, Sept. 1996;
####

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Prüfungsantrag gem. § 44 PatG ist gestellt

⑤④ Verfahren und Einrichtung zur analogen Detektion und Decodierung

⑤⑦ Die Erfindung betrifft Decodierverfahren für Block- und Faltungscodes, sowie Detektions- und Decodierverfahren für codierte Modulation, bei der im Gegensatz zum Stand der Technik nicht sequentielle Algorithmen und prozessorgesteuerte oder digital implementierte Decoder, sondern parallele, rückgekoppelte, lineare und nichtlineare Netzwerke zur Decodierung verwendet werden. Diese verarbeiten analoge Signale, und der Decoder akzeptiert, verarbeitet und liefert "Soft"-Werte. In vielen Fällen arbeitet der Decoder verzögerungsfrei und ist nur durch die Laufzeit und die Einschwingvorgänge in seiner Schnelligkeit begrenzt, welche durch die in der Schaltung vorhandenen parasitären Widerstände und Kondensatoren bestimmt sind. Besonders aufgebaute Codes erlauben die Decodierung mit Teilnetzwerken, die parallel, seriell oder verschachtelt konkateniert sind. Zwischen den Teilnetzwerken wird bei der Decodierung vorwärts und rückgekoppelt Information ausgetauscht. Die Erfindung ist anwendbar in allen digitalen Übermittlungs- und Speichersystemen, bei denen Kanalcodierung angewandt wird, insbesondere im Mobilfunk, Satellitenfunk und bei der leitungsgebundenen Übertragung.

DE 197 25 275 A 1

Seit der Frühzeit der Informations- und Codierungstheorie war es immer das Ziel, bei tolerierbarer Komplexität nahe an die von Shannon 1948 vorgegebenen Grenze zu kommen. Kanäle, für welche die Shannon Grenze leicht ermittelbar ist, sind: a) der Gaußsche Kanal (AWGN), näherungsweise realisiert bei Satellitenkanälen, b) der Rayleigh Kanal, näherungsweise realisiert bei der Schmalbandübertragung im Mobilfunk. Um den entsprechenden Shannongrenzen nahezukommen sollten praktische Codiervverfahren weiche ("soft") Werte und Kanalzustandsinformation (CSI) verwenden. In letzter Zeit sind iterative Decodiermethoden [HOP96] entwickelt worden, die es erlauben, sich der Shannongrenze relativ leicht zu nähern. Dabei werden sogenannte "Soft-in/Soft-out" Decoder [Hag92] verwendet, die nicht nur "Soft" Werte als Eingangswerte verwenden, sondern auch solche produzieren. Diese Verfahren arbeiten bisher algorithmisch und sequentiell in digitalen Prozessoren und verwenden z. B. den Viterbi-, den SOVA oder den Bahl-Algorithmus und daraus durch Vereinfachungen abgeleitete Algorithmen [HOP96].

Die vorliegende Erfindung geht noch einen Schritt weiter, es werden bei den Empfängern auch zur internen Signalverarbeitung "Soft"-Werte herangezogen, d. h. es werden durchwegs analoge (reelle) Signalwerte verwendet, die in einer Schaltung durch Strom- und Spannungswerte dargestellt sind. Dies stellt einen Schritt dar, der von der digitalen (binären) Welt zurück zu der analogen Welt führt. Der Empfänger wird realisiert als analoges, paralleles, nichtlineares Netzwerk, das mit den empfangenen Werten geladen wird und nach einem Einschwingvorgang die Ergebniswerte in analoger Form vorlegt. Das Vorzeichen dieser Ergebniswerte ist dann die Binärentscheidung, der Betrag ist dann die Zuverlässigkeit dieser Entscheidung.

Die Vorteile dieser neuen Methode liegen darin das praktisch ohne Verzögerung ("no latency") entschieden wird, keine Information verschwendet wird, hochparallel und integriert verarbeitet wird und "Soft"-Werte, d. h. Bits mit Zuverlässigkeit am Ausgang vorliegen. Natürlich können diese Netzwerke auch auf bisherigen sequentiellen Rechnern nachgebildet werden.

Grundlage der Erfindung

L-Werte und "Soft"-Bits

(K, N) Codes zur Fehlerkorrektur mit der Rate $R = K/N$ werden entweder durch die Generatormatrix G oder durch die Paritätsprüfmatrix H beschrieben [Fri95]. Die N-K Zeilen der Paritätsprüfmatrix geben die Prüfgleichungen, also z. B. für die m-te Zeile

$$\sum_{n=1}^{n=N} h_{m,n} \cdot x_n = 0,$$

oder nach x_k aufgelöst

$$x_k = \sum_{\substack{n=1 \\ n \neq k}}^{n=N} h_{m,n} \cdot x_n. \quad (1)$$

Addition \oplus und Multiplikation \odot sind im jeweiligen Galoisfeld auszuführen, also bei GF(2) in der bekannten modulo 2 Rechnung.

Die Bits kann man als abstrakte Elemente beliebig bezeichnen, also wie meist üblich mit 0,1 oder – wie hier verwendet – mit +1, –1. Man führt dann noch die Wahrscheinlichkeiten ein und die Loglikelihood-Verhältnisse

$$L(x) = \frac{\text{Prob}(x = +1)}{\text{Prob}(x = -1)}. \quad (2)$$

$L(x)$ ist eine reelle Zahl und die binäre (harte) Entscheidung ist

$$x = \text{sign}(L(x)) \quad (3)$$

und

$$|L(x)| \quad (4)$$

bedeutet die Zuverlässigkeit von x.

Mit

$$\lambda(x) = \tanh(L(x)/2) \quad (5)$$

bezeichnet man das sog. "Soft"-Bit, dessen Werte im Bereich von –1 bis +1 liegen.

Addiert man zwei statistisch unabhängige Bits (im GF(2) bzw. modulo 2)

$$\chi_3 = \chi_1 \oplus \chi_2 \quad (6) \quad 5$$

so gilt für die "soft" Bits [HOP96]

$$\lambda(\chi_3) = \lambda(\chi_1) \cdot \lambda(\chi_2), \quad (7) \quad 10$$

wobei die Multiplikation, die der reellen Zahlen ist. Für die L-Werte gilt dann die Beziehung

$$L(\chi_3) = 2 \cdot \operatorname{atanh}(\tanh(L(\chi_1)/2) \cdot \tanh(L(\chi_2)/2)), \quad (8)$$

die wir mit dem "Boxplus"-Symbol abkürzen: 15

$$L(\chi_3) = L(\chi_1) \boxplus L(\chi_2) \quad (9).$$

Diese Netzwerkelemente können wie in den Bildern 1, 2, 3 und 4 ausgeführt, auch als Bauteil realisiert werden.

Für die "Boxplus" Operation gilt die Näherung [HOP96]: 20

$$L(\chi_3) \approx \operatorname{sign}L(\chi_1) \cdot \operatorname{sign}L(\chi_2) \cdot \min(|L(\chi_1)|, |L(\chi_2)|) \quad (10).$$

Das Element \boxplus bildet ein wesentliches Bauteil in den nachfolgenden Realisierungen der Erfindung. Interessiert man sich beispielsweise für den L-Wert von χ_k in Gl. (1), so ergibt sich 25

$$L(\chi_k) = \sum_{\substack{n=1 \\ n \neq k \\ h_{m,n} \neq 0}}^{n=N} \boxplus L(\chi_n), \quad (11) \quad 30$$

wobei die Summe im Boxplus- Sinne über alle x zu nehmen ist bei denen $h_{m,n}$ von Null verschieden ist und der Index von k abweicht.

Falls man bei einer Implementierung die Multiplikation scheut, kann man auch in den Log- λ -Bereich Λ gehen: 35

$$\Lambda(\chi) = -\ln|\lambda| = -\ln \tanh(|L(\chi)|/2). \quad (12).$$

Umgekehrt erhält man

$$|\lambda| = e^{-\Lambda(\chi)}, \quad (13) \quad 40$$

$$|L(\chi)| = 2 \operatorname{atanh}(e^{-\Lambda(\chi)}). \quad (14).$$

Damit hat man die durch einfache Addition reeller positiver Zahlen auszuführende Betragsbeziehung

$$\Lambda(\chi_3) = \Lambda(\chi_1) + \Lambda(\chi_2), \quad (15) \quad 45$$

während für das Vorzeichen $\chi_i = \pm 1$ gilt

$$\chi_3 = \chi_1 \oplus \chi_2. \quad 50$$

Man beachte, daß beide Transformationen $|L|$ nach Λ und Λ nach $|L|$ nach der gleichen Funktion $f(w)$

$$f(w) = \ln \frac{1 + e^{-w}}{1 - e^{-w}} \quad (16) \quad 55$$

verlaufen. Als nichtlineare Bauteile sind sie im **Bild 2** dargestellt.

Übertragung von Bits

Nach der Übertragung des Bits χ über einen BSC oder einen Gaußschen/Fading Kanal, hat man den Empfangswert y und 60

$$L(x|y) = \log \frac{P(x = +1|y)}{P(x = -1|y)} = \log \left(\frac{p(y|x = +1)}{p(y|x = -1)} \cdot \frac{P(x = +1)}{P(x = -1)} \right). \quad (17) \quad 65$$

und mit den L-Werten

$$\begin{aligned}
 L(x|y) &= \log \frac{\exp(-\frac{E_s}{N_0}(y-a)^2)}{\exp(-\frac{E_s}{N_0}(y+a)^2)} + \log \frac{P(x=+1)}{P(x=-1)} \\
 &= L_c \cdot y + L(x).
 \end{aligned}
 \tag{18}$$

Dabei ist $L_c = 4a \cdot E_s/N_0$ für einen Fading Kanal mit der Amplitude a . Für den Gaußschen Kanal ist $a = 1$ und für den BSC ist L_c gleich

$$L_c = \log((1 - P_0)/P_0).$$

Deshalb wird L_c die Zuverlässigkeit oder Kanalzustandsinformation (CSI) des Kanals genannt.

Das Netzwerk wird gemäß der Erfindung mit den Werten $L(x|y)$ geladen, wobei die CSI und die a priori Information bekannt sein müssen und, wie im Bild 5 gezeigt, gewichtet werden. Ist $L(x)$ nicht bekannt, so wird es zu Null gesetzt.

Bausteine und Aufbau des Decodiernetzwerkes

Codes können auch durch Graphen beschrieben werden [TAN81], [FOR96]. Ausgehend von den Paritäts- oder Codegleichungen enthält das erfindungsgemäße Decodiernetzwerk verschiedene Elemente die in den Abbildungen dargestellt sind:

- (a) Element Kanalgewichtung
- (b) Element L nach λ und Element λ nach L
- (c) Element L nach Λ und Element Λ nach L
- (d) Element binäre Addition und entsprechende Elemente in λ (Multiplikation), L (Boxplus) und Λ (Addition)
- (e) Ausführungsbeispiel Boxplus
- (f) Element Λ -Addition und Näherung der Boxplus-Operation getrennt nach Vorzeichen und Betrag.

Ein "Kreis"- oder "Boxplus"-Element ohne gepfeilte Linien bedeutet, daß jede Linie doppelt ist, also aus dem bidirektionalen Element Signale heraus- und hereingehen.

Ausführungsbeispiele mit zwei Linien sind in den Abb. 7 und 6 gezeigt. Jedes Bit des Codes wird durch einen Kreis, jede Prüfgleichung durch ein "Boxplus"-Element dargestellt. Auf diese Weise wird das Netzwerk konstruiert und mit den gewichteten empfangenen Kanalwerten geladen. Es läuft dann ein eventuell durch das Eigenrauschen getriggelter Ausgleichsvorgang ab. Nach einer frei wählbaren Zeit werden die "Soft Outputs" der gewünschten Bits abgegriffen.

Die Operationen des Netzwerkes können im L -, λ - oder Λ -Bereich durchgeführt werden. Dabei ist darauf zu achten, daß so wenig nichtlineare Operationen wie möglich durchgeführt werden. Die nichtlinearen Transformationen können auch durch Tafeln realisiert werden. Die Implementierung des Netzwerkes kann auf verschiedene Art erfolgen:

- Realisierung mit diskreten Bauelementen
- Integration analoger Bausteine und ihrer Verknüpfungen
- Realisierung durch programmierbare Prozessoren und durch integrierte digitale Schaltungen

Ausführungsbeispiele für Decodiernetzwerke für binäre Codes

Blockcodes

Wiederholungscode $(N, 1, N)$

Bei einem Wiederholungscode der Länge N ist der extrinsische Ergebniswert:

$$L_e = \sum_{n=2}^N L(x_i|y_i)$$

und der "Soft-Output"

$$L(x_1|y_1) + L_e$$

Paritätsprüfcode SCPC $(N, N-1, 2)$

Bei einem SPC-Code der Länge N ist der extrinsische Ergebniswert:

$$L_{ei} = \sum_{\substack{n=1 \\ n \neq i}}^N \boxplus L(x_n | y_n)$$

5

und der "Soft-Output"

$$L(x_i | y_i) + L_{ei}$$

10

Ein Beispiel für einen (3, 2, 2) SPC Code ist in der **Abb. 7** gezeigt.

Hammingcode (7, 4, 3)

Der Hammingcode hat die im **Abb. 8** angegebene Prüfmatrix H . Die drei Prüfgleichungen sind durch die 3 "Boxplus"-Elemente repräsentiert, an denen die entsprechenden Informationsbits u und Prüfbits p inzident sind. Für diese Bits werden in das Netzwerk die entsprechenden Kanalwerte $L(x|y)$ geladen und nach der Einschwingzeit werden die Ausgangswerte u abgelesen. Die Boxpluselemente haben, wie oben, beschrieben 4 Eingänge und 3 Ausgänge, letzteres an den ungepfleiteten Linien.

15

Eingebettete Codes: Reed-Muller-Codes

Eingebettete Codes, die zum Beispiel durch Summenkonstruktionen [Fri95] erzeugt werden, können auch durch eingebettete Netzwerke decodiert werden. Eine solche Codefamilie sind die Reed-Muller $RM(r, m)$ Codes mit

20

$$(2^m, \sum_{i=0}^r \binom{m}{i}, 2^{m-r})$$

Sie können rekursiv folgendermaßen erzeugt werden: Mit $u \in RM(r+1, m)$ und $v \in RM(r, m)$ ist

25

$\{u \oplus v\} \in RM(r+1, m+1)$.

Weiterhin ist $RM(0, m)$ ein Wiederholungscode und $RM(m-1, m)$ ein Parity-Check-Code.

Wie in **Abb. 9** gezeigt, läßt sich der RM-Code rekursiv aufbauen. Entsprechend läßt sich das Decodiernetzwerk rekursiv aufbauen und sukzessive auf die bekannten Netzwerke des Wiederholungscodes (Kreis-Element) und des SCPC-Codes ("Boxplus"-Element) zurückführen. Über die Verbindungslinien der einzelnen Teilnetzwerke wird wieder extrinsische Information ausgetauscht.

30

Faltungscodes

35

Einfachstes Beispiel

Das einfachste Decodiernetzwerk wird in **Abb. 10** durch einen systematischen Faltungscodes vom Gedächtnis 1 und Rate 1/2 generiert. Der Faltungscodes ist terminiert, entweder dadurch, daß das Startbit und das Schlußbit u_5 zu 0 gesetzt werden, oder das letzte Bit u_5 als Startbit verwendet wird ("Tailbiting"). Im ersten Fall entsteht ein (10,4)-Code, im zweiten Fall der höherrangige und quasizyklische (10,5)-Code. Das zyklische Decodiernetzwerk zeigt **Abb. 10**. Im ersten Fall wird u_5 durch $+\infty$ geladen. Im zweiten Fall wird bei der Codierung der Speicher durch u_5 vorbelegt. Bei der Decodierung werden dann die L -Werte aller 10 Bits in Form der Kanalwerte $L(x|y)$ an das Netzwerk gelegt.

40

Faltungscodes in nichtsystematischer Form

Die **Abb. 11** zeigt einen Faltungscodes mit Gedächtnis 2 in "Tailbiting Form" und das zugehörige Decodiernetzwerk. Geladen wird das Netzwerk an den "Boxplus" Elementen, ausgelesen wird an den Kreis-Elementen. Das Netzwerk kann, wie es **Abb. 12** zeigt, im Prinzip beliebig weit ausgedehnt werden, mit ∞ -Werten abgeschlossen oder zyklisch terminiert werden. Das Decodiernetzwerk ist zyklisch zu schließen, wenn der Code in "Tailbiting"-Form ist. Die ersten und letzten u -Werte sind $\pm\infty$ zu setzen, wenn der Coder durch bekannte Bits initiiert und terminiert ist verwendet man Codes mit höherem Gedächtnis, so erhöhen sich im Gegensatz zur Trellisdarstellung die Zahl der Knoten nicht, lediglich die Zahl der Verbindungen steigt. Die Zahl der an den beiden "Boxplus"-Elementen anliegenden Verbindungen ist eins mehr als die Zahl der Einsen in den Zeilen der Generatormatrix, also in obigem Beispiel wegen

50

$$G = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

4 bzw. 3 Verbindungen. Beim Gedächtnis 4 Code

55

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \end{pmatrix}$$

5

sind es 4 bzw. 5 Verbindungen.

Auch eine Punktierung [Fri95] zur Erzielung hochratiger Codes ist leicht möglich, indem die entsprechenden "Box-plus"-Elemente weggelassen werden. Für eine Punktierung mit der Matrix $\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$, die aus dem Rate 1/2 einen 2/3-Code erzeugt wird jedes 2. "Boxplus"-Element weggelassen.

10

Rückgekoppelte Faltungscodes in systematischer Form

Für rückgekoppelt erzeugte systematische Faltungscodes sieht, wie in **Abb. 13** gezeigt, das Decodiernetzwerk etwas anders aus. Die Zustandsvariablen a_i werden neu eingeführt und das Ergebnis liegt an den äußeren Kreis-Elementen an, welche die 2. Parity Elemente ersetzt haben.

15

Auch hier läßt sich, wie in **Abb. 14**, ein ausgedehntes Netzwerk erzeugen.

Realisierung des Netzwerkes durch einen Prozessor

Obwohl das Hauptanliegen der Erfindung die Realisierung durch analoge Schaltungen ist, kann auch eine mehr konventionelle prozessororientierte Implementierung durchgeführt werden. Dazu werden die Operationen taktweise ausgeführt und die Ausgänge über einen Zwischenspeicher und die nächste Verknüpfungsoperation weitergegeben. Beispielsweise würden dann in **Abb. 12** die nichtbepfeilten Linien, die ja bidirektional sind, ihre Ausgangswerte an ein Zwischenspeicherelement weitergeben. Wird beispielsweise von rechts ein neues Codebitpaar angelegt, werden alle Ausgänge über diese Speicher um einen Takt nach links weitergereicht. Auch die Codebits in **Abb. 14** würden dann über einen Zwischenspeicher um einen Takt weitergereicht. Mit dieser Anordnung lassen sich neue einlaufende Bits und ihre L-Werte sequentiell verarbeiten.

25

Parallel- und seriell verketteter Codes

30

Die parallele Verkettung wurde bei den sog. "Turbo"-Codes wurde in [BG79] durch iterative sequentielle Verfahren decodiert.

35

Die **Abb. 15** zeigt eine solche Verkettung, bei der die einzelnen Teilcodes über den Interleaver $\Pi^{(1)}$ die Informationsbits nochmals verwürfelt erhalten und die Paritätsbits $p^{(1)}$ erzeugen. Auf der Decodierseite wird erfindungsgemäß der Teilcode durch sein Netzwerk dargestellt und diese Netzwerkeile durch die Verbindungen V verbunden (Die benötigten Deinterleaver sind der Einfachheit halber in der Zeichnung weggelassen). Über diese Verbindungen V werden die extrinsischen Informationen der einzelnen Decodiernetzwerke ausgetauscht.

40

Entsprechendes gilt für eine serielle Verkettung der Codes wie unten im **Abb. 15** dargestellt. Der Decoder des inneren Code C_i gibt seine extrinsische Information an den äußeren Decoder DEC_o , der wiederum seine Information an DEC_i zurückführt. Auch in diesen verketteten nichtlinearen Netzwerken laufen Einschwingvorgänge ab, deren Zeitkonstante nur durch die parasitären Widerstände und Kondensatoren bestimmt sind, da die oben beschriebenen Netzwerkbaulemente verzögerungsfrei sind.

Parallelverkettete Blockcodes

45

Die **Abb. 16** zeigt ein Beispiel eines parallel verketteten (8,4,4)-Codes, der aus (3,2,2)-SCPC-Teilcodes besteht. Gezeigt ist auch das Ergebnis der ersten Iteration, die durch horizontales Auftrennen des verketteten Netzwerkes entstehen würde. Erfindungsgemäß wird jedoch das Netzwerk nicht aufgetrennt, sondern mit den 8 Kanalwerten (4 an "Boxplus"-4 an "Kreis"-Elementen) geladen. Nach Abklingen des Ausgleichsvorganges werden an den 4 "Kreis"-Elementen die Ergebniswerte abgelesen.

50

Parallelverkettete Faltungscodes

55

Die **Abb. 17** zeigt ein Ausführungsbeispiel eines parallelverketteten Faltungscodes. Jeder der 2 Teildecoder ist ein Netzwerk eines systematischen Decoders. Die extrinsischen Teilinformationen werden über die Verbindungsmatrix V mit den durch die Interleavermatrix und die Permutationen Π festgelegten Verbindungen ausgetauscht. Bei der Decodierung werden die L-Werte die zu $\{u_i\}$, zu $\{p_i^{(1)}\}$, sowie zu $\{p_i^{(2)}\}$

Faltungscode, der seriell mit Blockcodes verkettet ist

60

Abb. 18 zeigt als Beispiel einen (2,1) "Tailbiting"-Faltungscode mit Gedächtnis 1, der seriell mit zwei (3,2,2)-SCPC-Blockcodes verkettet ist. Über den Verbindungsring werden innerer Code C_i und die äußeren Codes C_o so vernetzt, daß die darüber ausgetauschten inneren und äußeren extrinsischen Informationen möglichst statistisch unabhängig sind.

65

Ausführungsbeispiele zu Decodiernetzwerke für mehrstufige codierte Modulation

Mehrstufige codierte Modulation nach Imai und Hirakawa kann auch mehrstufig decodiert werden. Dabei werden vorteilhaft Soft-Werte und eine iterative Rückführung verwendet [WH93]. Erfindungsgemäß wird nun auch diese Decodie-

rung durch ein Netzwerk durchgeführt, wobei die Teilnetzwerke der Codes der einzelnen Stufen ihre Ergebnisse zur Metrikberechnung aus den empfangenen QAM-(MPSK)-Werten zurückkoppeln. Das Verfahren wird hier durch eine 8-PSK-Modulation mit $M=3$ Stufen erklärt, jedoch ist das Verfahren für jede stufen-codierte Modulation anwendbar.

Die 8-PSK Signalmenge wird wie in der Abb. 19 gezeigt durch sog. "Setpartitioning" in Teilmengen aufgespalten, wobei jede Teilmengenpartitionierung $i = 1, 2, 3$ durch einen Code geschützt wird mit der gleichen Länge N und den Bits $b^{(1)}, b^{(2)}, b^{(3)}$. 5

Die Decodierung erfolgt nach Abb. 20 durch einen Mehrstufendecoder DECI, bei dem der Code jeder Stufe durch ein Decodiernetzwerk decodiert wird. Es kann aber auch jeder andere zu den Codes passender Soft-in/Soft-out-Decoder verwendet werden. Das Ergebnis jeder Decodierstufe wird rückgekoppelt und in der Metrikberechnung als a priori Information verwendet. Die Metrikberechnung erfolgt für die i -te Stufe nach folgender Formel 10

$$L(x^{(i)}|y) = \log \frac{\sum_{x \neq x^{(i)}} \exp(-\frac{E_s}{N_0} d_0^2 + \sum_{j \neq i} x^{(j)} L(x^{(j)})/2)}{\sum_{x \neq x^{(i)}} \exp(-\frac{E_s}{N_0} d_1^2 + \sum_{j \neq i} x^{(j)} L(x^{(j)})/2)}$$

Dabei sind 15

- d_0 Die euklidische Distanz von y zum Signalpunkt mit $\chi^{(i)} = 0(+1)$
- d_1 Die euklidische Distanz von y zum Signalpunkt mit $\chi^{(i)} = 1(-1)$
- $L(\chi^{(i)})$ Die rückgekoppelte a priori Information über das Bit $\chi^{(i)}$

20

Es kann folgende Näherung für die Metrik verwendet werden:

$$L(x^{(i)}|y) \approx \max_{x \neq x^{(i)}} (-\frac{E_s}{N_0} d_0^2 + \sum_{j \neq i} x^{(j)} L(x^{(j)})/2) - \max_{x \neq x^{(i)}} (-\frac{E_s}{N_0} d_1^2 + \sum_{j \neq i} x^{(j)} L(x^{(j)})/2)$$

25

Diese Näherung läßt sich für 8-PSK, wie in Abb. 21 gezeigt, im Metriknetzwerk ausführen.

Für 4-PSK ist die entsprechende Näherung und das Metriknetzwerk in Abb. 22 gezeigt. Es lassen sich aber auch – wie oben angegeben – Realisierungen im A-Bereich angeben, bei der Log-L Werte und demzufolge Summationen verwendet werden können. 30

Das Decodiernetzwerk wird dann mit den $M \cdot N$, im Beispiel mit $3 \cdot M$ empfangen und gewichteten Werten geladen und über das Metrik-Netzwerk den Decodern zugeführt.

Das einfachste Beispiel verwendet zu 8-PSK die Codes WH (8, 1, 8), SPC (8, 7, 2) und uncodiert (8, 8, 1) und erzielt 3 dB asymptotischen Gewinn (Fri95). 35

Der erste Decoder ist durch ein "Kreis"-Element, der zweite durch ein "Boxplus"-Element verwirklicht, der 3. Decoder entfällt.

Die $M = 3$ Code-Decodiernetzwerke liefern dann ohne Verzögerung die Apriori-Werte, die wiederum über das Metrik-Netzwerk rückgekoppelt in die M Decodiernetzwerke einfließen. Nach dem Einschwingvorgang werden am Ausgang der Decodiernetzwerke die "Soft-Output" Werte der Informationsbits abgegriffen. 40

Abbildungsverzeichnis

- 1 Nichtlineares Element zur Transformation von L nach λ und λ nach L
- 2 Nichtlineares Element L nach Λ und Element Λ nach L 45
- 3 Element binäre Addition (mod 2 bzw. XOR-Addition) und die entsprechenden Elemente in λ (Multiplikation), L (Boxplus) und Λ (Addition)
- 4 Ausführungsbeispiel Boxplus-Addition
- 5 Element Kanalgewichtung: Erzeugung der Eingangswerte des Netzwerkes aus den empfangenen Werten y , der Kanalzustandsinformation L_c und der a priori Information $L(x)$ 50
- 6 GF(2) bzw. mod 2 Verknüpfung dreier Bits und die zugehörige bidirektionale "Boxplus"-Operation. Linien ohne Pfeile bedeuten bidirektionale Operation. Beispiel: $L(u_1) = L(u_2) \square + L(p) = +1.0 \boxplus -1.5 \approx -1.0$
- 7 Detailliertes Ausführungsbeispiel bidirektionaler Bauteile
- 8 Aus der Paritätsprüfmatrix H entwickeltes Decodiernetzwerk eines Hamming Codes $(N, K, d) = (7, 4, 3)$
- 9 Beispiel zur Kombination von Untercode-Netzwerken: Konstruktion von Reed-Muller Codes 55
- 10 Einfacher Faltungscodex mit Rate 1/2, Gedächtnis 1, mit 5 Informationsbits und sein Decodiernetzwerk. Der Code ist in "Tailbiting"-Form
- 11 Nichtsystematischer Faltungscodex mit Rate 1/2, Gedächtnis 2, mit 5 Informationsbits und sein Decodiernetzwerk. Der Code ist in "Tailbiting"-Form
- 12 Nichtsystematischer Faltungscodex mit Rate 1/2, Gedächtnis 2, mit beliebig vielen Informationsbits 60
- 13 Beispiel eines rückgekoppelten systematischen Faltungscodes in "tailbiting" Form mit 6 Informationsbits
- 14 Systematischer Faltungscodex mit Rate 1/2, Gedächtnis 2, in rückgekoppelter Form mit beliebig vielen Informationsbits.
- Das Decodiernetzwerk ist zyklisch zu schließen, wenn der Code in "Tailbiting"-Form ist
- 15 Serielle und Parallele Verkettung von Netzwerken 65
- 16 Parallel verketteter Code und sein Decodiernetzwerk
- 17 Beispiel parallel verketteter Faltungscodes mit systematischem Komponentencode nach Bild 8 und sein Decodiernetzwerk. Die Teilnetzwerke sind durch eine Permutationsverbindungsbrücke miteinander verbunden, über welche die

extrinsischen Ausgangswerte der "Kreis"-Bauteile ausgetauscht werden

- 18 Beispiel seriell verketteter Codes mit innerem (12,6)-Gedächtnis 1-Komponentencode, wie in Bild 4, aber mit 6 Informationsbits. Der äußere Code besteht aus 2 äußeren SCPC Codes mit den Parametern (3,2,1). Der Gesamtcode ist ein (12,4)-Code. Das Decodiernetzwerk besteht hier aus einem inneren und zwei äußeren Teilnetzwerken. Innere und äußere Teilnetzwerke sind durch eine Permutationsverbindungsbrücke miteinander verbunden, über die die extrinsischen Ausgangswerte ausgetauscht werden. Die Permutation soll die höchstmögliche statistische Unabhängigkeit der zu einem äußeren Codewort gehörenden Bits sicherstellen
- 19 "Set Partitioning" bei einem 8-PSK-Signal
- 20 Decodiernetzwerk zur Decodierung von codierter 8-PSK Modulation
- 21 Metrik Netzwerk bei 8-PSK codierter Modulation
- 22 Metrik Netzwerk bei 4-PSK codierter Modulation

Literatur

- 15 [Fri95] B. Friedrichs, "Kanalcodierung", Springer Verlag, Berlin, 1995.
- [HOP96] J. Hagenauer, E. Offer, L. Papke, "Iterative decoding of binary block and convolutional codes", IEEE Trans. on Inf. Theory, vol. IT-42, pp 429-425, March 1996.
- [Hag92] J. Hagenauer, Soft-In/Soft-Out: The benefits of using soft decisions in all stages of digital receivers", in Proc. Uni Int. Workshop on DSP Techniques applied to Space Communications, ESTEC Noordwijk, The Netherlands, Sept. 1992.
- 20 [BGT93] C. Berrou et al., "Near Shannon limit error-correcting and decoding: Turbo-Codes (1)", Proc. IEEE International Conference on Communication (ICC), Geneva, Switzerland, pp. 1064-1070, May 1993.
- [LOD93] J. Lodge et al., "Separable MAP "filters" for the decoding of product and concatenated codes," Proc. IEEE International Conference on Communication (ICC), Geneva, Switzerland, pp. 1740-1745, May 1993.
- 25 [WH93] Th. Woerz, J. Hagenauer, "Decoding of M-PSK-Multilevel Codes," European Transactions on Telecommunications ETT, Vol 4, No. 3, pp 299-308, 1993.
- [TAN81] R.M. Tanner, "A recursive approach to low complexity codes," IEEE Transactions on Information Theory, vol. IT-27, pp 533-547, Sep. 1981.
- [FOR96] G. D. Forney, "The forward-backward algorithm", Proc. of the 1996 Allerton Conference, Allerton, Illinois, Sep. 1996.

Patentsprüche

1. Verfahren zur Decodierung von Block- und Faltungscodes, **dadurch gekennzeichnet**, daß der Decoder aus einem nichtlinearen Netzwerk besteht, das aus den Code- oder Paritätsgleichungen abgeleitet wird, indem alle in einer Gleichung inzidenten Bits oder Symbole durch ihre Loglikelihood-Werte als reelle Größen dargestellt werden. Weiterhin werden in dem Netzwerk alle durch die Code- oder Paritätsgleichungen verbundenen Symbole durch die das oben beschriebene sog. "Boxplus" Element verbunden, während alle Symbole durch das oben beschriebene Speicher "Kreis"- Element gespeichert und dargestellt werden. Diese Größen können als Ströme, Ladungen oder Spannungen, aber auch als reelle Fest- oder Fließkommavariablen, z. B. in einem Prozessor oder einem Schaltkreis dargestellt werden. Neben diesen Bausteinen sind im Decodiernetzwerk oder im Prozessor Speicher, Umsetzungstabellen, und Nichtlinearitäten vorhanden. Die Nichtlinearitäten sind typischerweise Tangenshyperbolicus- und inverse Tangenshyperbolicus- Funktionen. Das Netzwerk wird durch die Kanal-Loglikelihood-Werte $L(\chi|y) = L_c y + L(\chi)$ geladen, wobei L_c die Kanalzustandsinformation, y den empfangenen Kanalwert z. B. den matched Filter Ausgang, L_c die Kanalzustandsinformation CSI und $L(\chi)$ eine mögliche a priori Information über das Symbol χ darstellen. Das Netzwerk kann im Rückkopplungszweig durch Filter beliebiger Ordnung in seinem Einschwingverhalten beeinflusst werden. Nach Abklingen des Einschwingvorganges werden die codierten oder die Informationsbits ausgelesen.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die Tanh-Nichtlinearität, bzw. ihre Inverse, im "Boxplus" Element durch eine andere Sigmoid- Funktion angenähert wird, insbesondere durch die Näherung

$$L(\chi_3) \approx \text{sign}L(\chi_1) \cdot \text{sign}L(\chi_2) \cdot \min(|L(\chi_1)|, |L(\chi_2)|)$$

- oder dadurch gekennzeichnet, daß die "Boxplus" Operation im Log-Bereich durchgeführt wird und dadurch als additives Element realisiert wird:

$$\Lambda(\chi) = -\ln |\lambda| = -\ln \tanh |L(\chi)|/2.$$

Die Umkehroperation ist

$$L(\chi) = 2 \tanh(e^{-\Lambda(\chi)}).$$

Die Addition wird durch die Betragsbeziehung

$$\Lambda(\chi_3) = \Lambda(\chi_1) + \Lambda(\chi_2),$$

realisiert, das Vorzeichen $\chi_i = \pm 1$ durch die XOR-Verknüpfung

$$\chi_3 = \chi_1 \oplus \chi_2.$$

3. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß der Decoder auf einen Faltungscode mit Abschluß oder auf einen Faltungscode mit zyklischem Abschluß angewandt wird, derart daß die bekannten (Tail)-Bits mit $\pm\infty$, bzw. dem höchstmöglichen Wert im Schaltkreis oder Prozessor bewertet werden oder beim zyklischen Abschluß der Decoder eine Ringstruktur hat. Bei einem zyklischen Faltungscode ("tail-biting") werden in bekannter Weise die letzten M Bits des Blockes in das Gedächtnis M des Encoders geladen.

4. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß der Decoder auf zwei oder mehr parallel verkettete Codes angewandt wird, welche dieselbe Information, die jedoch umgeordnet ("interleaved") ist, mehrfach codieren. Dementsprechend sind zwei oder mehr Basisnetzwerke als Decoder zu verwenden. Erfindungsgemäß wird nun zwischen den beiden Decodernetzwerken die verschachtelte ("interleaved") extrinsische Information ausgetauscht, wie im Bild 17 beschrieben.

5. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß das Decodernetzwerk auf seriell verkettete Codes angewandt wird. Dementsprechend sind zwei oder mehr Basisnetzwerke als Decoder zu verwenden. Erfindungsgemäß wird nun zwischen den inneren und äußeren Decodernetzwerken die "interleaved" extrinsische Information der codierten Bits ausgetauscht, wie in den Abb. 15 und 18 beschrieben.

6. Verfahren und Decodernetzwerk nach Anspruch 1, dadurch gekennzeichnet, daß das Decodernetzwerk auf eine mehrstufige codierte Modulation (QAM) angewandt wird. Als Codiervorgang kann jedes codierte Modulationsverfahren verwendet werden, das in bekannter Weise die QAM Signalmenge durch sog. "Setpartitioning" in Teilmengen aufspaltet, wobei jede Teilmengenpartitionierung durch einen Code geschützt wird. Erfindungsgemäß erfolgt nun die Decodierung durch einen Mehrstufendecoder, bei dem der Code jeder Stufe durch ein Decodernetzwerk nach Anspruch 1-6 decodiert wird. Es kann aber auch jeder andere zu den Codes passender Soft-in/Soft-out Decoder verwendet werden. Erfindungsgemäß wird nun weiterhin das Ergebnis jeder Decodierstufe rückgekoppelt und in der Metrikberechnung als a priori Information verwendet. Die Metrikberechnung erfolgt für die i-te Stufe nach folgender Formel

$$L(x^{(i)}|y) = \log \frac{\sum_{x \neq x^{(i)}} \exp(-\frac{E_s}{N_0} d_0^2 + \sum_{j \neq i} x^{(j)} L(x^{(j)})/2)}{\sum_{x \neq x^{(i)}} \exp(-\frac{E_s}{N_0} d_1^2 + \sum_{j \neq i} x^{(j)} L(x^{(j)})/2)}$$

Dabei sind

- d_0 Die euklidische Distanz von y zum Signalpunkt mit $\chi^{(i)} = 0(+1)$
- d_1 Die euklidische Distanz von y zum Signalpunkt mit $\chi^{(i)} = 1(-1)$
- $L(\chi^{(i)})$ Die rückgekoppelte a priori Information über das Bit $\chi^{(i)}$

Fernerhin kann erfindungsgemäß die folgende Näherung für die Metrik verwendet werden:

$$L(x^{(i)}|y) \approx \max_{x \neq x^{(i)}} (-\frac{E_s}{N_0} d_0^2 + \sum_{j \neq i} x^{(j)} L(x^{(j)})/2) - \max_{x \neq x^{(i)}} (-\frac{E_s}{N_0} d_1^2 + \sum_{j \neq i} x^{(j)} L(x^{(j)})/2)$$

Erfindungsgemäß können die Berechnungen auch in einem Metriknetzwerk durchgeführt werden oder in vorab adressierbaren (ROM)-Tafeln gespeichert werden.

7. Verfahren nach den Ansprüchen 1 bis 6, dadurch gekennzeichnet, daß das Decodernetzwerk durch Prozessoren oder digitale sequentielle Schaltkreise realisiert wird. In diesem Fall werden die Operationen taktweise ausgeführt und die Ausgänge über Zwischenspeicher and die nächste Verknüpfungsoperation weitergegeben. Auch die einlaufenden Codebits werden dann über einen Zwischenspeicher um einen Takt weitergereicht. Mit dieser variierten Anordnung lassen sich neue einlaufende Bits und ihre L-Werte sequentiell verarbeiten.

Hierzu 19 Seite(n) Zeichnungen

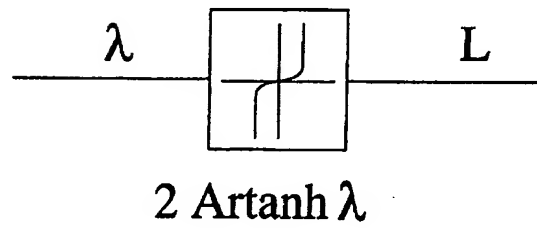
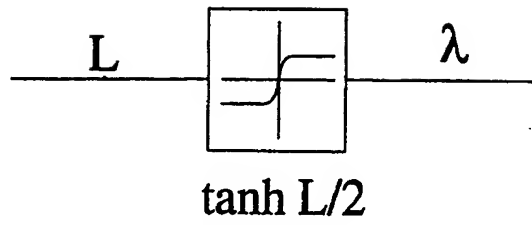
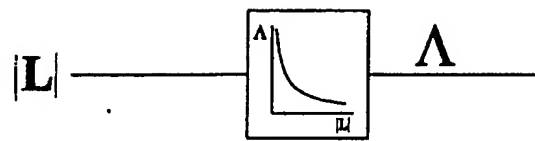
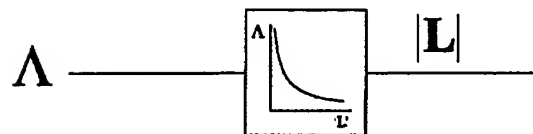


Abbildung 1: Nichtlineares Element zur Transformation von L nach λ und λ nach L



$$-\ln \tanh \left| \frac{L(x)}{2} \right|$$



$$2 \operatorname{atanh} \left(e^{-\Lambda(x)} \right)$$

Abbildung 2: Nichtlineares Element L nach Λ und Element Λ nach L

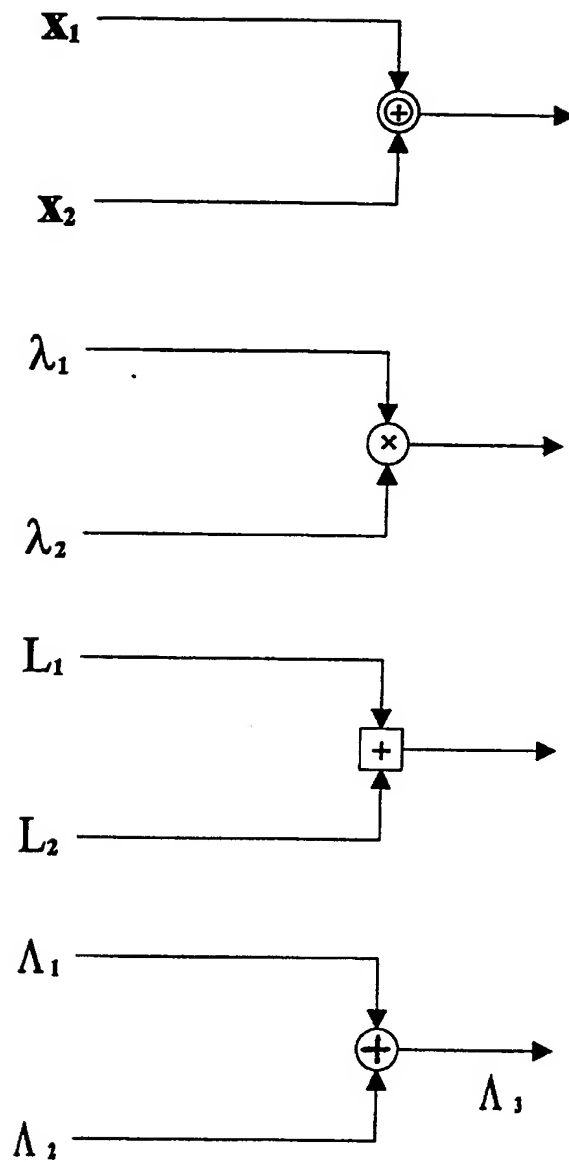


Abbildung 3: Element binäre Addition (mod 2 bzw XOR - Addition) und die entsprechenden Elemente in λ (Multiplikation), L (Boxplus) und Λ (Addition)

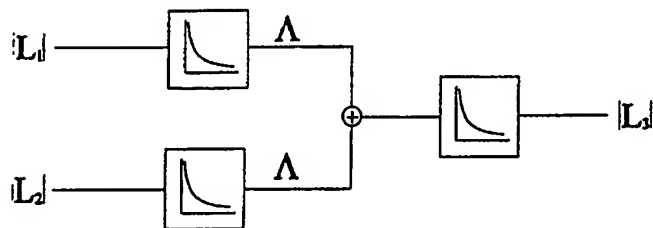
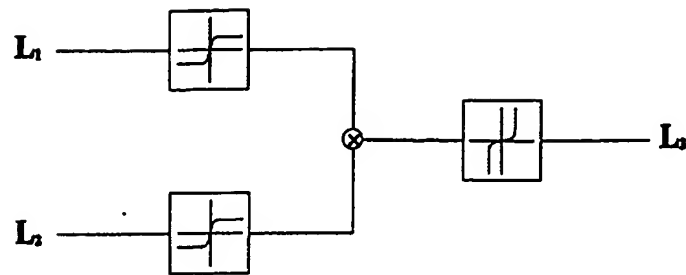


Abbildung 4: Ausführungsbeispiel Boxplus-Addition

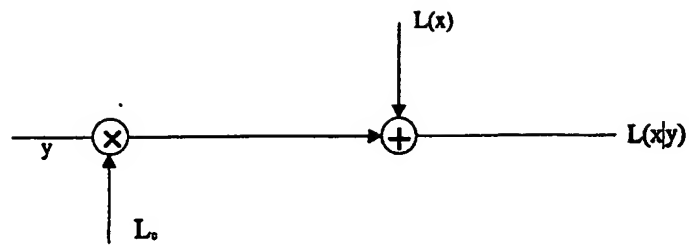


Abbildung 5: Element Kanalgewichtung: Erzeugung der Eingangswerte des Netzwerkes aus den empfangenen Werten y , der Kanalzustandsinformation L_c und der a priori Information $L(x)$.

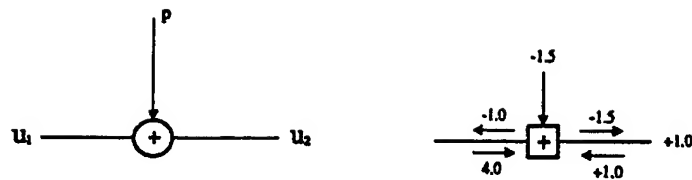


Abbildung 6: GF(2) bzw. mod 2 Verknüpfung dreier Bits und die zugehörige bidirektionale "Boxplus"-Operation. Linien ohne Pfeile bedeuten bidirektionale Operation. Beispiel: $L(u_1) = L(u_2) \boxplus L(p) = +1.0 \boxplus -1.5 \approx -1.0$

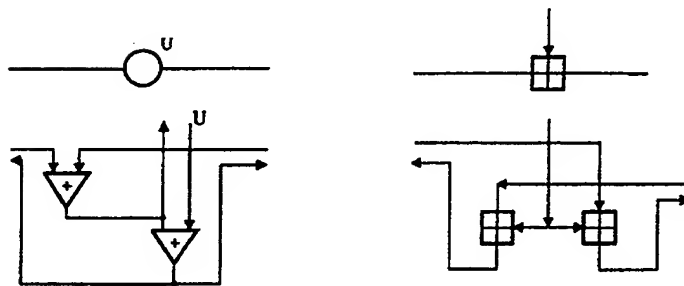


Abbildung 7: Detailliertes Ausführungsbeispiel bidirektionaler Bauteile.

$$\begin{array}{ccccccc}
 & u_1 & u_2 & u_3 & u_4 & p_1 & p_2 & p_3 \\
 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\
 \mathbf{H} = & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
 & 0 & 1 & 1 & 1 & 0 & 0 & 1
 \end{array}$$

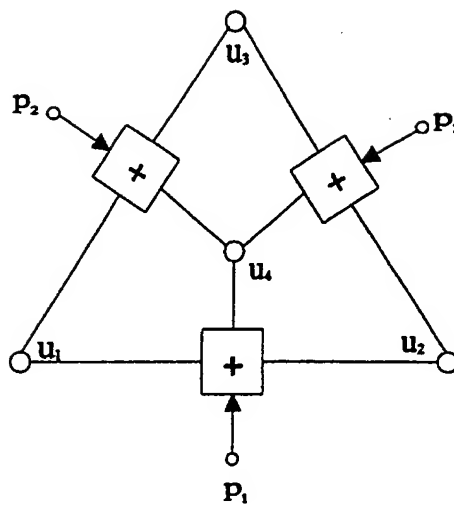
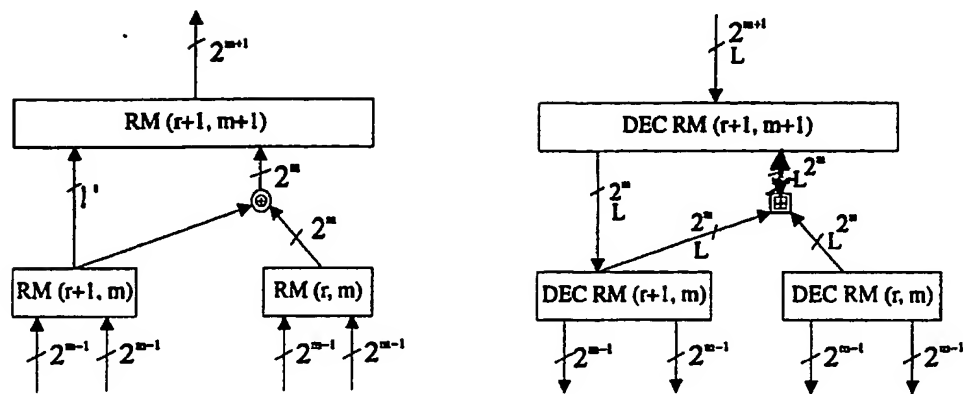


Abbildung 8: Aus der Paritätsprüfmatrix H entwickeltes Decodiernetzwerk eines Hamming Codes $(N,K,d) = (7,4,3)$



\oplus , \boxplus means componentwise operation

Abbildung 9: Beispiel zur Kombination von Untercode-Netzwerken: Konstruktion von Reed-Muller Codes.

CONVOLUTIONAL CODE

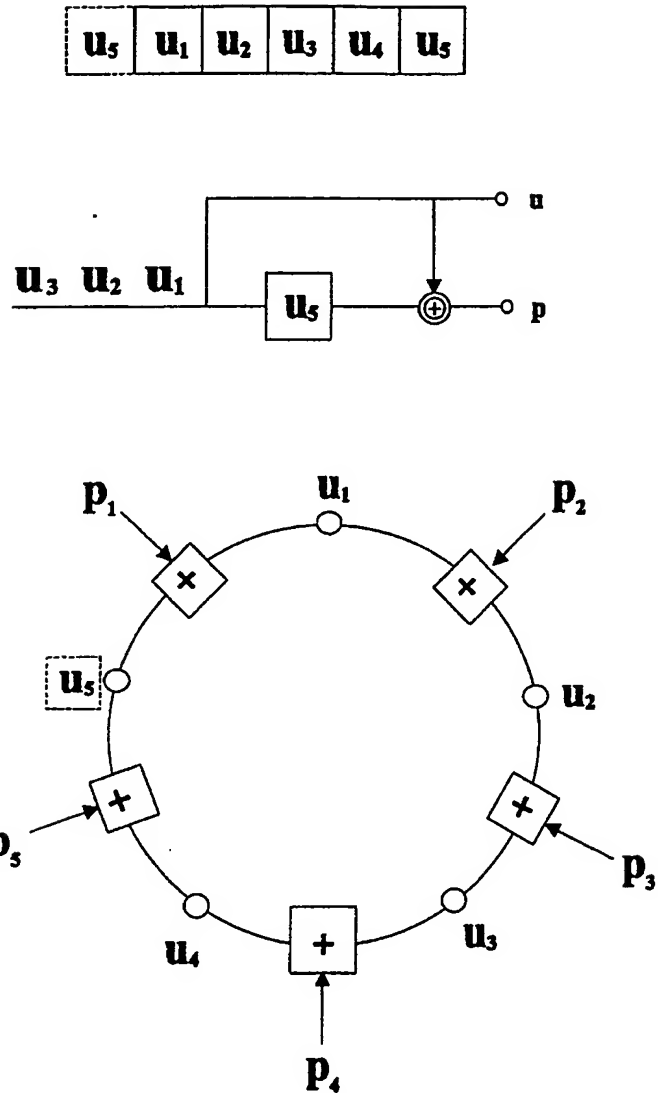


Abbildung 10: Einfacher Faltungscodier mit Rate 1/2, Gedächtnis 1, mit 5 Informationsbits und sein Decodiernetzwerk. Der Code ist in "Tailbiting"-Form.

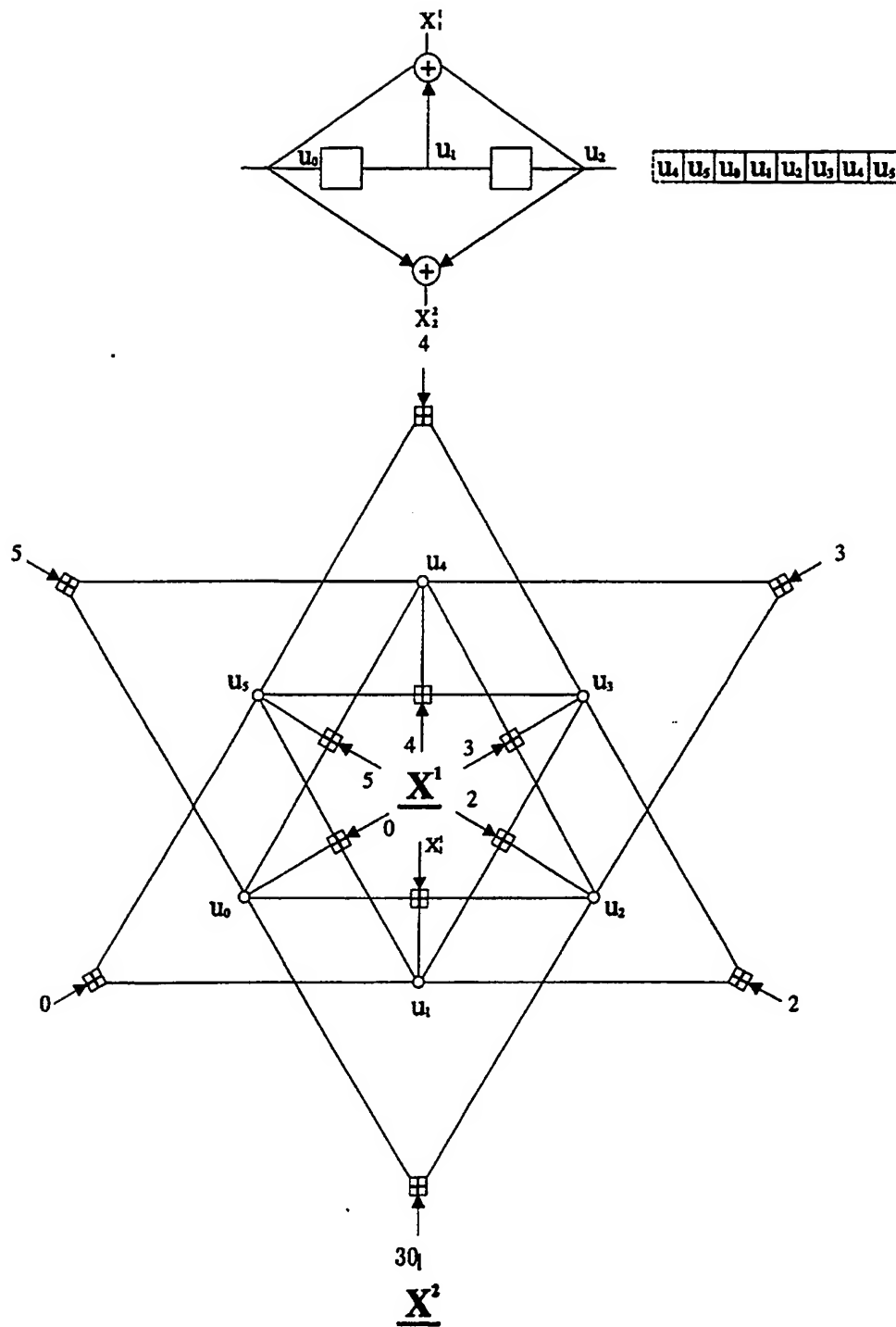


Abbildung 11: Nichtsystematischer Faltungscode mit Rate 1/2, Gedächtnis 2, mit 5 Informationsbits und sein Decodiernetzwerk. Der Code ist in "Tailbiting" Form.

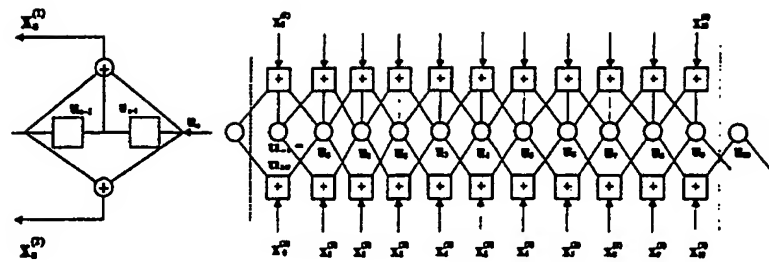


Abbildung 12: Nichtsystematischer Faltungscode mit Rate 1/2, Gedächtnis 2, mit beliebig vielen Informationsbits.

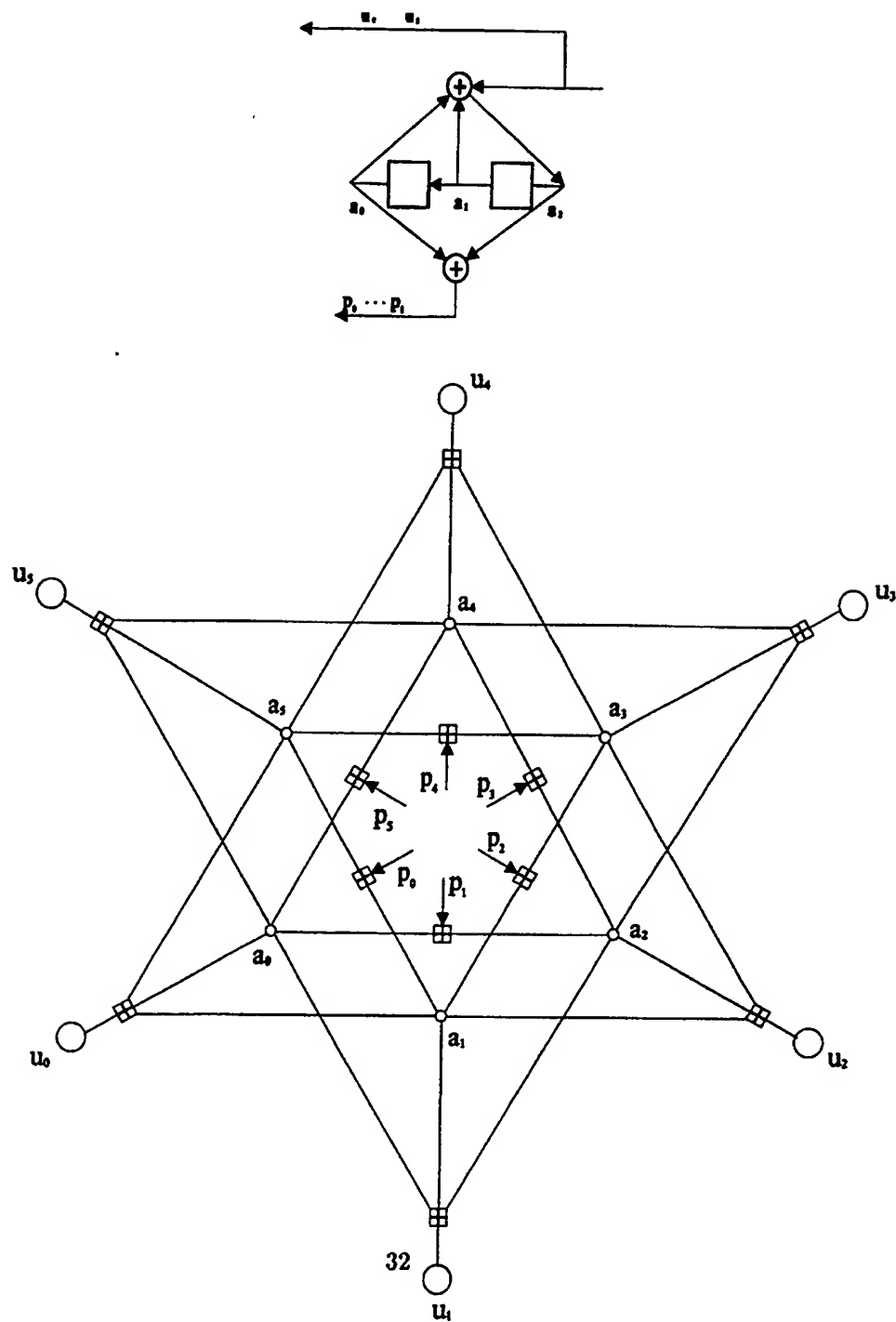


Abbildung 13: Beispiel eines rückgekoppelten systematischen Faltungscodes in "tailbiting" Form mit 6 Informationsbits.

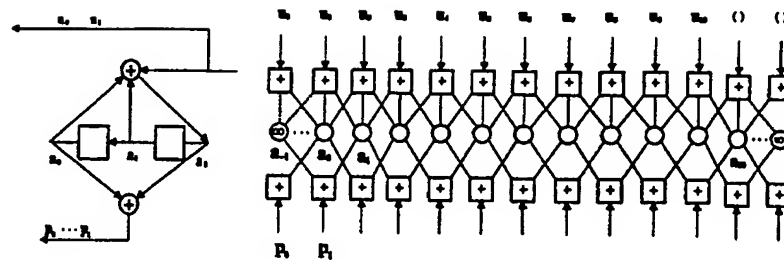


Abbildung 14: Systematischer Faltungscode mit Rate 1/2, Gedächtnis 2, in rückgekoppelter Form mit beliebig vielen Informationsbits. Das Decodieretzwerk ist zyklisch zu schließen, wenn der Code in "Tailbiting"-Form ist.

Prinzip der a) parallelen und b) seriellen Verkettung

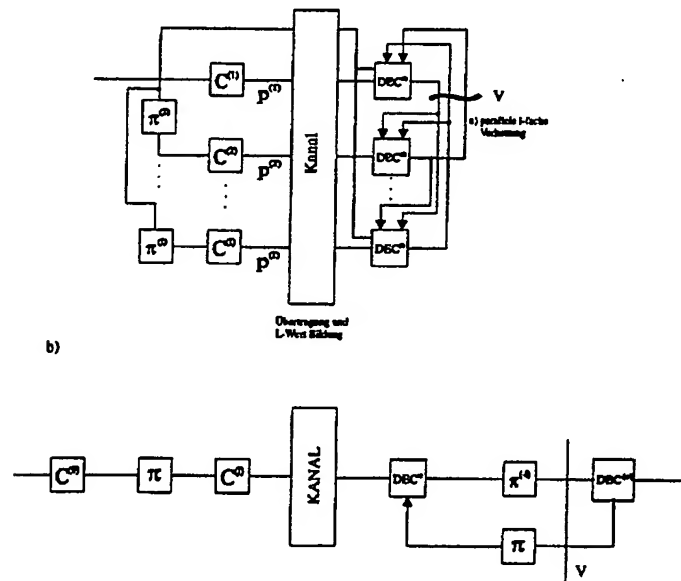


Abbildung 15: Serielle und Parallele Verkettung von Netzwerken.

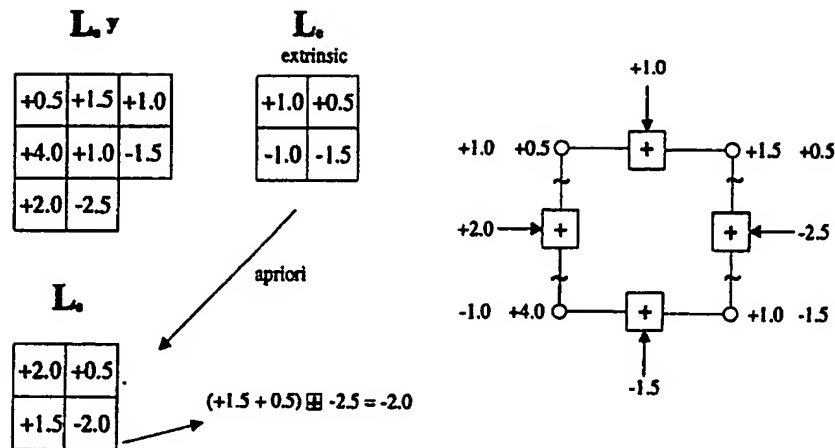


Abbildung 16: Parallel verketteter Code und sein Decodiernetzwerk

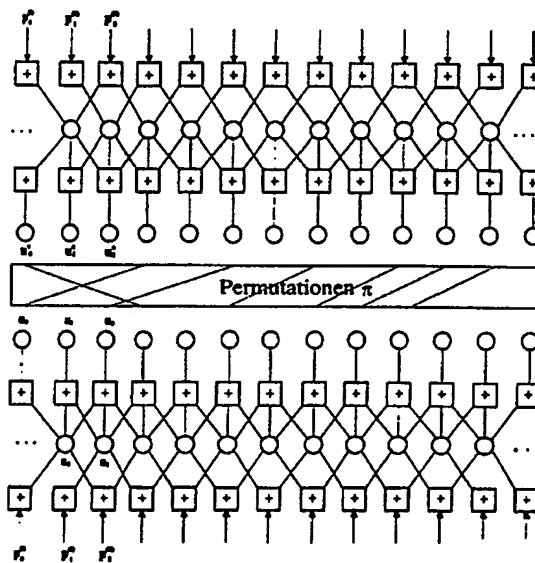


Abbildung 17: Beispiel parallel verketteter Faltungscodes mit systematischem Komponentencode nach Bild 8 und sein Decodiernetzwerk. Die Teilnetzwerke sind durch eine Permutationsverbindungsbrücke miteinander verbunden, über welche die extrinsischen Ausgangswerte der "Kreis"-Bauteile ausgetauscht werden.

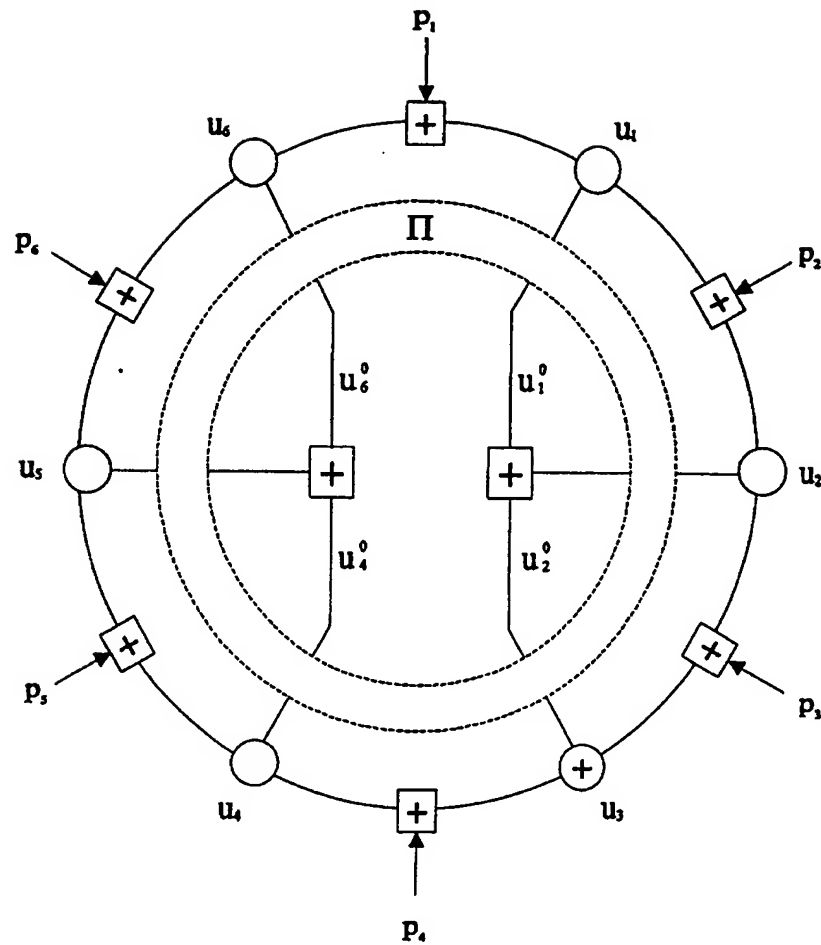


Abbildung 18: Beispiel seriell verketteter Codes mit innerem (12,6)-Gedächtnis 1-Komponentencode, wie in Bild 4, aber mit 6 Informationsbits. Der äußere Code besteht aus 2 äußeren SCPC Codes mit den Parametern (3,2,1). Der Gesamtcode ist ein (12,4)-Code. Das Decodiernetzwerk besteht hier aus einem inneren und zwei äußeren Teilnetzwerken. Innere und äußeren Teilnetzwerke sind durch eine Permutationsverbindungsbrücke miteinander verbunden, über die die extrinsischen Ausgangswerte ausgetauscht werden. Die Permutation soll die höchstmögliche statistische Unabhängigkeit der zu einem äußeren Codewort gehörenden Bits sicherstellen. .

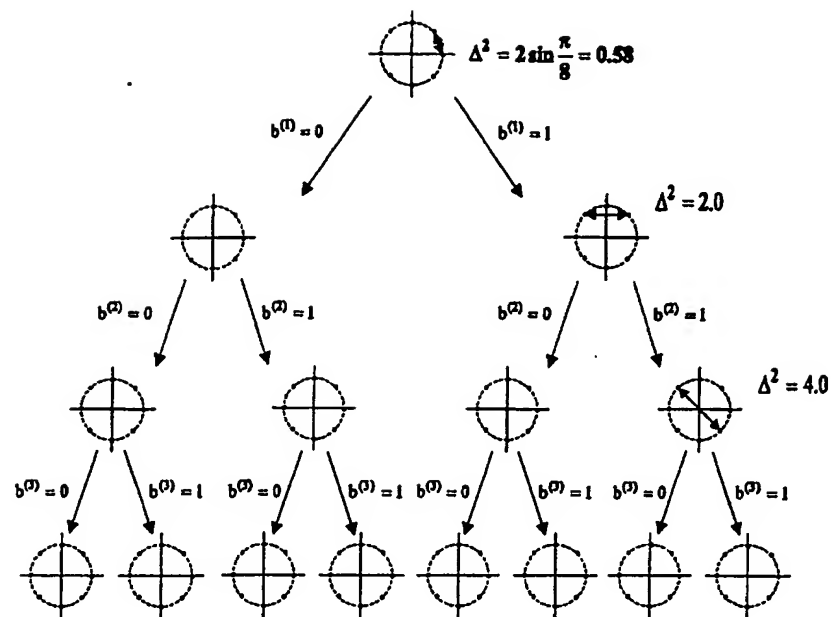


Abbildung 19: "Set Partitioning" bei einem 8-PSK-Signal.

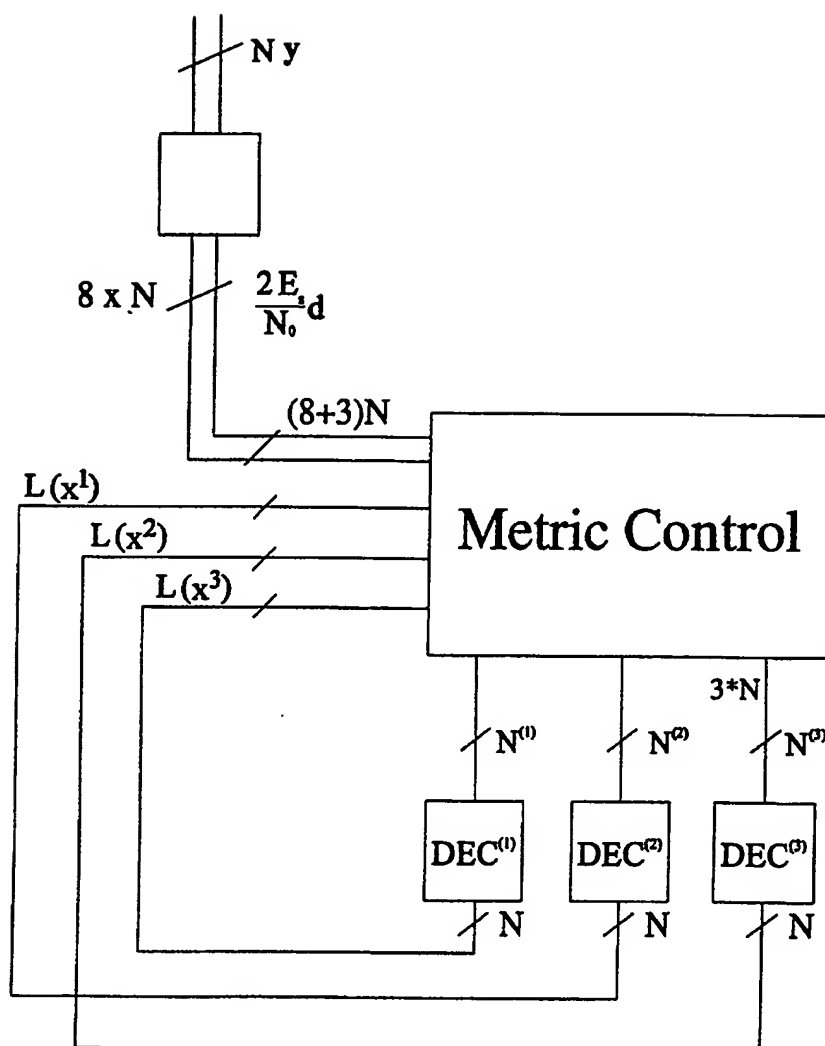
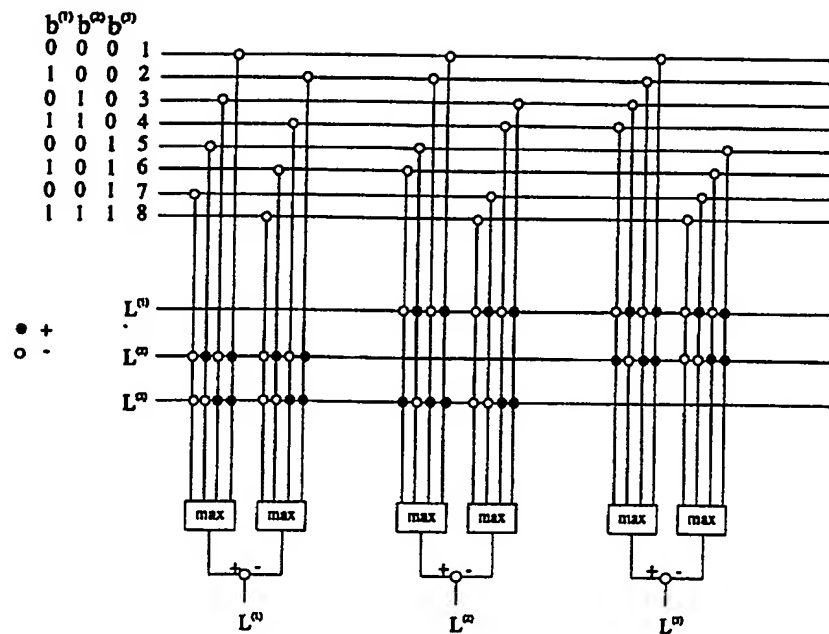


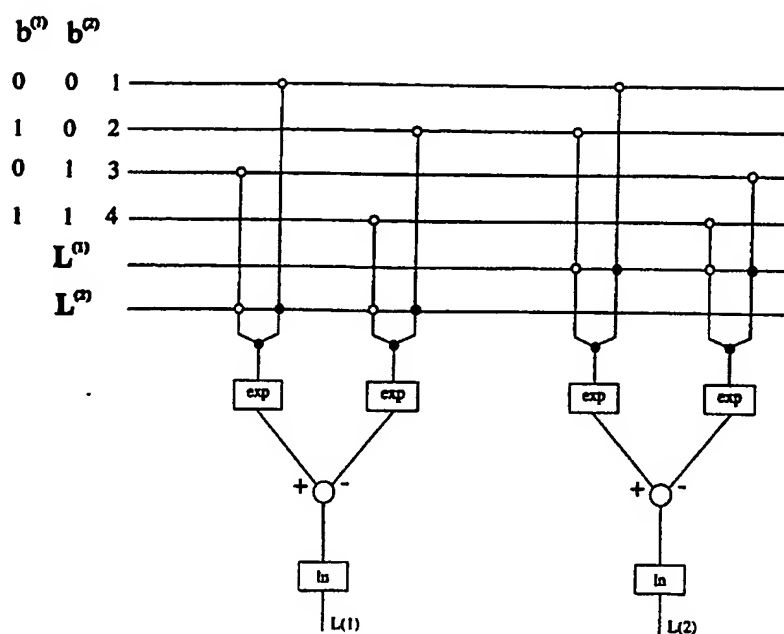
Abbildung 20: Decodiernetzwerk zur Decodierung von codierter 8-PSK Modulation.



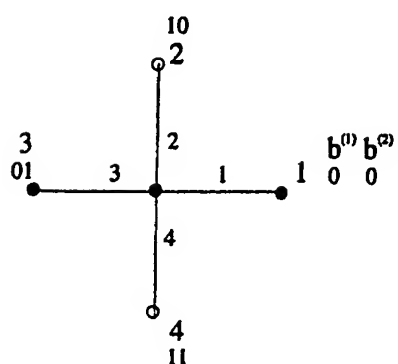
Inzidenzmatrix

1			-									
2												
3			-									
4												
5			-									
6												
7			-									
8												
9												
10	-	+	-	+	-	+	-	+	-	+	-	+
11	-	+	+	+	-	+	+	+	-	+	+	+

Abbildung 21: Metrik Netzwerk bei 8-PSK codierter Modulation.



4-PSK



Inzidenzmatrix

-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-

Abbildung 22: Metrik Netzwerk bei 4-PSK codierter Modulation.